

REMARKS

This is in response to the Office Action mailed on August 3, 2007 in which claims 1-3, 5, and 14-20 were rejected. With this Amendment, claims 1 and 14-18 are amended. Claim 19 is canceled and claims 4 and 6-13 were previously canceled. All amendments are fully supported by the specification and drawings. No new matter is added. Claims 1-3, 5, 14-18, and 20 are pending in this application. In light of the foregoing amendments and following remarks, Applicant respectfully requests advancement of this application to allowance.

Rejections Under 35 U.S.C. § 112

In the Office Action, claims 1, and 15-17 were rejected under 35 U.S.C. § 112. Applicant thanks the Examiner for noting these informalities in the claims. Claim 1 was rejected for the phrase “of the type” (line 1) lacking antecedent basis. Claim 1 is amended herein to recite “of a type.” Claims 15-17 were rejected as being method claims that depend from a system claim. Claims 15-17 are amended herein to properly recite “system” rather than method. With these amendments, withdrawal of the rejections under 35 U.S.C. § 112 is respectfully requested.

The amendments to the claims under this section were not made to overcome an art based rejection. Therefore, such amendment should not be considered in a limiting manner.

Rejections Under 35 U.S.C. § 102

In the Office Action, claims 1-3, 5 and 14-20 were rejected under 35 U.S.C. § 102(b) as being anticipated by the Eder patent application (U.S. Patent Application 2001/0034628). Applicant respectfully traverses the rejection, and the correctness of the rejection is not conceded. Nevertheless, in the interest of moving this application into condition for allowance, claims 1, 14, and 18 are amended. As noted above, claims 15-17 are also amended to correct informalities. The Eder patent does not disclose the limitations recited in Applicant’s independent claims 1, 14, and 18.

A. Overview of the Present Disclosure

In general terms, this patent application discloses a consulting system that allows a user to receive advice about a particular topic across a network. One example described in the application assists a user with understanding legal questions across a network, such as to advise the user as to whether a defect in a product was a breach of contract. To adequately advise the

user, the system asks the user one or more questions—such as “is the defect in the product that you ordered a ‘hidden defect’?” (See, e.g., table 2, page 13.) The system receives answers to the questions and eventually provides the user with a final answer.

As described in the background of the present application, a problem in the prior art was that experts in a particular field (e.g., law) often do not have the necessary programming skills to define processes and database operations necessary to answer users’ questions across a network. Similarly, those experienced in programming typically lack the knowledge of a particular field (e.g., law) to prepare questions and algorithms necessary to advise users on questions relating to the particular field.

In one aspect, the present patent application describes a system and method that allows an expert to easily define the questions and algorithms necessary to provide users with appropriate advice without requiring the expert to perform the complex programming. One of the ways that this is accomplished is by organizing data into objects including Problem objects and Unit objects.

A problem object typically includes a question to be asked of the user and also the possible answers that the user can select from. An example question of one problem object is shown as Q1 in table 2 (page 13). The problem object includes a question “Is a defect in the product that you ordered a ‘hidden defect’?” The problem object also includes the possible answers of “Yes” or “No”. Other example questions and answers include Q2-Q5. Problem objects can include an execution condition. The execution condition defines the conditions on which the problem is performed. (See, “condition” on page 9.)

A unit object typically includes one or more of the problem objects. The unit object also includes rules that define the logical flow between problem objects (also referred to as rules) and an answer script. The answer script defines the final answers to be presented to the user based on the user’s answers to the problem objects. Unit objects can include a compulsory condition. A compulsory condition includes a designation of a problem object or another unit object that is to be performed prior to the “condition” of the problem unit. In this way, the compulsory condition enables re-use of other problem or unit objects. (See, “Compulsory condition” on page 10.)

B. FIG. 5A of the Eder Patent Application Generally

Eder describes a system and method for evaluating the probable impact of user-specified or system generated changes in business value drivers on the components of value, revenue, expense and capital change, of a commercial enterprise. See Abstract.

The Office Action refers to FIG. 5A and paragraph 0047 of Eder as disclosing the elements of the claims. FIG. 5A illustrates processing that is completed by a portion of application software (200) that extracts, aggregates and stores the information required for system operation from: the basic financial system database (10), operation management system database (15), advanced financial system database (30), sales management system database (35), human resource information system database (40), external databases found on the internet (5) and the user (20). Paragraph 0062.

The process of FIG. 5A includes software blocks 201-212. The process begins at Start (block 201) and then prompts the user to enter system settings (block 202). Data dictionary and database summary information are extracted from various databases (block 203). Block 204 determines whether the current calculation is a comparison to a prior valuation or if it is a stand-alone calculation (para. 0081). If "stand alone", then blocks 205 to 207 are preformed. Alternatively, if a "comparison", then block 209 is performed. Blocks 205 to 207 prompt the user for input. Block 205 prompts the user for any input that is required to define data fields for the extracted data dictionaries and the data dictionary of the application software (para. 0082). Block 206 prompts the user to specify the account number segment or segments that will be used to define the enterprise being valued by the system (para. 0084). Block 207 prompts the user (20) via a component definition data window (904) to specify the account segment or segments that will be used to define the expense and capital sub-components for each enterprise (para. 0086). Block 209 checks to determine if all of the available financial data have been included in a revenue, expense, or capital component or sub-component (para. 0089). If so, block 210 is performed. If not, block 208 is performed. Block 208 displays a screen that provides the user with the ability to redefine previously stored definitions to include unassigned financial data. The revised definitions are saved and block 209 is again performed. Block 210 is then performed to retrieve the debit or credit balances from various databases and stored. Block 212 determines if any data is missing in required periods, and if so block 213 is performed to prompt the user for the missing data. If no data is missing, processing continues as shown on FIG. 5B.

C. Claims 1, 14, and 18

Claim 1 recites a consulting system of a type which provides a consulting service of a specific field to a user across a network. The consulting system includes a database system and a service component. The database system includes a plurality of problem objects and a plurality of unit objects. Each of the problem objects has an identifier, an execution condition, questions which will be given to the user, and answers which will be selected by the user as properties of the problem objects. Each unit object has an answer script for reflecting a result of a series of processes and including at least one of the plurality of problem objects, the plurality of unit objects including a first unit object, wherein rules between the objects are defined depending on corresponding properties of the objects, and wherein the first unit object includes as a property a compulsory condition which precedes the execution condition of the plurality of problem objects. The service component provides an interface screen for the consulting service to the user via the network, and supplies personalized final answers to the user by performing the answer script if the user answers the questions supplied by the rules, wherein the service component executes the plurality of problem objects in a sequence according to the identifiers of the problem objects which have been input as a property of the first unit object, and wherein the execution condition determines whether the problem object is executed or not.

Claim 14 recites a computer readable medium containing computer executable instructions which when executed by a computer perform a method of providing consulting services of a specified field to a user. The consulting service includes a consulting system and a network. The method comprises defining rules between a plurality of objects depending on corresponding properties of the objects, the plurality of objects including a plurality of problem objects and a plurality of unit objects; storing the plurality of problem objects in a database system, each of the problem objects having an identifier, an execution condition, questions which will be given to the user, and answers which will be selected by the user as properties of the problem objects; storing the plurality of unit objects in the database system, each of the unit objects having an answer script for reflecting a result of a series of processes and including at least one of the plurality of problem objects, the plurality of unit objects including a first unit object wherein the first unit object includes as a property a compulsory condition which precedes the execution condition of the plurality of problem objects; executing the plurality of problem objects in a sequence according to the identifiers of the problem objects which have been

inputted as a property of the first unit object, unless an execution condition has not been satisfied; and providing an interface screen from a service component to the user via the network, the interface screen supplying personalized final answers to the user by performing the answer script if the user answers questions of the problem objects defined in the rules.

Claim 18 recites a method similar to the method of claim 14 noted above.

Eder fails to disclose a service component that executes the plurality of problem objects in a sequence according to the identifiers of the problem objects which have been input as a property of the first unit object, as recited in claim 1. Eder also fails to disclose executing the plurality of problem objects in a sequence according to the identifiers of the problem objects which have been inputted as a property of the first unit object, unless an execution condition has not been satisfied, as recited in claims 14 and 18.

In the Response to Arguments, the Office Action states that the questions (e.g., 205-208) are executed in a sequence. However, Eder does not disclose executing the plurality of problem objects in a sequence according to the identifiers of the problem objects which have been input as a property of the first unit object. Furthermore, Eder does not disclose a first unit object having a property that includes identifiers of problem objects.

Rather, Eder shows in FIG. 5A the software algorithm that is executed. Nowhere does this software algorithm refer to identifiers of problem objects to determine the sequence of execution. Rather, the software algorithm itself defines a single execution process as shown.

Eder also fails to disclose **a first unit object including as a property a compulsory condition which precedes the execution condition of the plurality of problem objects**, as recited in claims 1, 14, and 18. As described on page 10 of the specification, a compulsory condition is a property of a unit object that includes a designation of a problem object or another unit object that is to be performed prior to the “condition” of the problem unit. The compulsory condition enables re-use of other problem or unit objects.

The Office Action asserts that block 204, shown in FIG. 5A of Eder is a compulsory condition. However, block 204 is a software decision block that determines whether flow should proceed in one direction or another. In contrast, the compulsory condition is **a property of the first unit object** that refers to one or more problem objects or another unit object. When used, the property allows re-use of existing problem objects or unit objects.

In light of the foregoing, claims 1, 14, and 18 are in condition for allowance.

D. Claims 2-3, 15-17, and 19-20

Claim 19 has been canceled herein. Claims 2-3 and 5 ultimately depend from claim 1. Claims 15-17 ultimately depend from claim 14. Claim 20 ultimately depends from claim 18. Claims 1, 14, and 18 are in condition for allowance, as discussed above. Therefore, claims 2-3, 5, 15-17, and 20 are also allowable.

In addition, the Eder patent, whether taken alone or in combination with the knowledge of one skilled in the art, does not teach or suggest all of the elements recited in independent claims 1, 14, and 18.

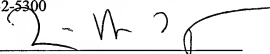
Conclusion

In view of the above amendments and remarks, Applicant respectfully requests a Notice of Allowance. There may be additional reasons that the pending subject matter is patentably distinct from the cited references in addition to those discussed herein. Applicant reserves the right to raise any such arguments in the future. If the Examiner believes a telephone conference would advance the prosecution of this application, the Examiner is invited to telephone the undersigned at the below-listed telephone number.

Respectfully submitted,

MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
(612) 332-5300

Date: January 3, 2008



Name: Brian H. Batzli
Reg. No.: 32,960
BHB/BAT:ae